Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Non-Monotonic Logical Modeling for Regulatory Genetic Networks

A. Rocca[1], E.Fanchon[1], L. Trilling[1]

[1]Lab. TIMC-IMAG, U. de Grenoble, France

13-03-2017

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency r
On going works

Motivations

To show the interest of a **declarative** approach based on **Answer Set Programming** (ASP)

- for modeling Thomas' logical discrete **Genetic Regulatory Networks** (GRNs)
- for inducing GRNs a priori consistent with experiments (**reverse engineering**).
- for taking into account both automatic inconsistency repairing and gene interaction properties which are only **generally true**, by using **default rules** provided by ASP.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Declarative approach. Personnal wiew

Four steps methodology.

- Formalization (network structure, behaviors, ...) with constraints.
- Consistency test (see below).
- Extraction of properties (theorems).
- Choice of experiments / Knowledge addition. Return to step1.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency r
On going works

Content

1. Basics: ASP and logical modeling of Thomas GRNs

2. Applications

3. Generally true Additivity Constraints and automatic consistency repairing

4. On going works

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Belief revision with ASP (Answer Set Programming).

ASP is a logic programming technology based on a non monotonic logic with models said **stable** which are minimal. Rules are:

$$a_0 : -a_1, \ldots, a_m, not\ a_{m+1}, \ldots, not\ a_n$$

The typical example for introducing to non monotonic logics:

- From the axioms in ordinary (monotonic) logic:
  $flies(X) \Leftarrow bird(X)$
  $bird(tweety)$
  one deduces $flies(tweety)$.

- The problem is with penguins. Taking them into account demands:
  - completing the 1st axiom by $\neg penguin(X)$ as a premise,
  - qualifying **by hand** every bird (is it or not a penguin ?).

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Advantages of ASP. Belief revision(cont.)

With ASP, these manual revisions may be avoided by using
defaults. Unless the contrary is **proved**, a bird is not a penguin.
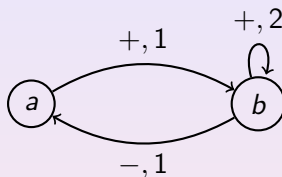
- From:
  ```
  flies(X) :- bird(X), not penguin(X).
  bird(tweety).
  ```
  one deduces flies(tweety).
- If by addition of new knowledge (e.g. result of
  experimentation), penguin(tweety) can been proved then
  flies(tweety) is no more deducible (non monotony).

Additivity constraints on gene interactions (see later) can be
modeled by such defaults.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Thomas GRNs. Interaction graph

Focal equations:

$$X_a = \begin{cases} K_a & \text{if } x_b < \theta_b^1 \\ K_a^b & \text{if } x_b \geq \theta_b^1 \end{cases}$$

$$X_b = \begin{cases} K_b & \text{if } x_a < \theta_a^1 \text{ and } x_b < \theta_b^2 \\ K_b^a & \text{if } x_a \geq \theta_a^1 \text{ and } x_b < \theta_b^2 \\ K_b^b & \text{if } x_a < \theta_a^1 \text{ and } x_b \geq \theta_b^2 \\ K_b^{ab} & \text{if } x_a \geq \theta_a^1 \text{ and } x_b \geq \theta_b^2 \end{cases}$$

x_a: (discrete) concentration of protein a. θ_a^1: threshold of a.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Thomas GRNs. Dynamics

- *Focal* equations relate a state $[x_a, x_b]$ and its focal state $[X_a, X_b]$ indicating in which direction are its neighboring successors, thanks to parameters K.

- Semantics of signs:

  Observability constraint (**always** true) for $a \xrightarrow{+,1} b$:
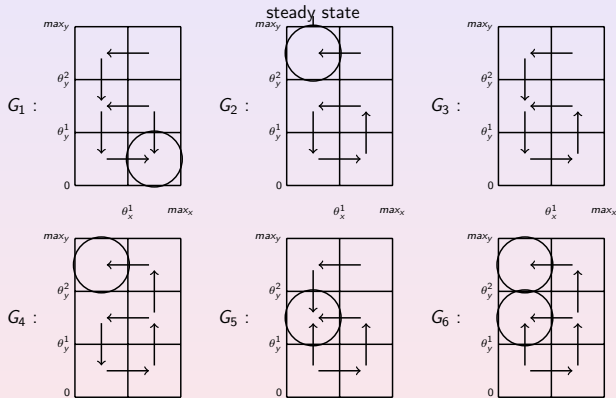 $(K_b < K_b^a) \vee (K_b^b < K_b^{ab})$
  i.e activation in at least on case.
  Additivity constraint (**generally** true) for $a \xrightarrow{+,1} b$:
  $(K_b \leq K_b^a) \wedge (K_b^b \leq K_b^{ab})$
  i.e. no inhibition.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Transition graph

Transition graphs satisfying observability and additivity constraints.
One equilibrium for G_1, G_3, G_5. **Mutistationarity** for G_2, G_4, G_6.

There are $2^2 * 3^4 = 332$ possible set of parameters.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Experiments (behaviors) represented as paths

Examples:

- Enforcing the existence of a path of two successive identical states (steady state) gives all transition graphs except G_3.
- Enforcing the existence of a path beginning with the state $[0, 0]$ and reaching the state w $[0, 2]$ leads to the models G_4 and G_6.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency r
On going works

Other facilities

Include

- automatic inconsistency repairing.

- mutant specification.

- minimization (interactions and thresholds values): the ASP software provides para-logical operators like `#minimize{f_1,...,f_n}` that produces only models with the lowest number of literals `f_i` true.

- deduction of properties on domains specified by biologists : for example, $(K_b < K_b^a) \vee \neg(K_b < K_b^{ab})$ true in all models.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency r
On going works

Content

1. Basics: ASP and logical modeling of Thomas GRNs

2. **Applications**

3. Generally true Additivity Constraints and automatic consistency repairing

4. On going works

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Carbon Starvation in *E.Coli* [Ropers et al., Biosyst., 2006]
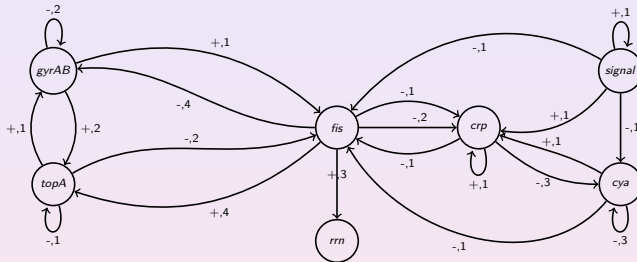
Two **steady states**: 1) with a high concentration of Fis and a high supercoiling (e.g. high ratio GyrAB / TopA), 2) after carbon deprivation, with a high concentration of Crp and a weaker supercoiling.

Two **response paths** to the two stresses: carbon deprivation, carbon-source availability.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Inconsistency repairing. [Corbin et al., Biosyst., 2009]

Inconsistency with: observability and additivity constraints + constraints enforcing steady states and response paths.
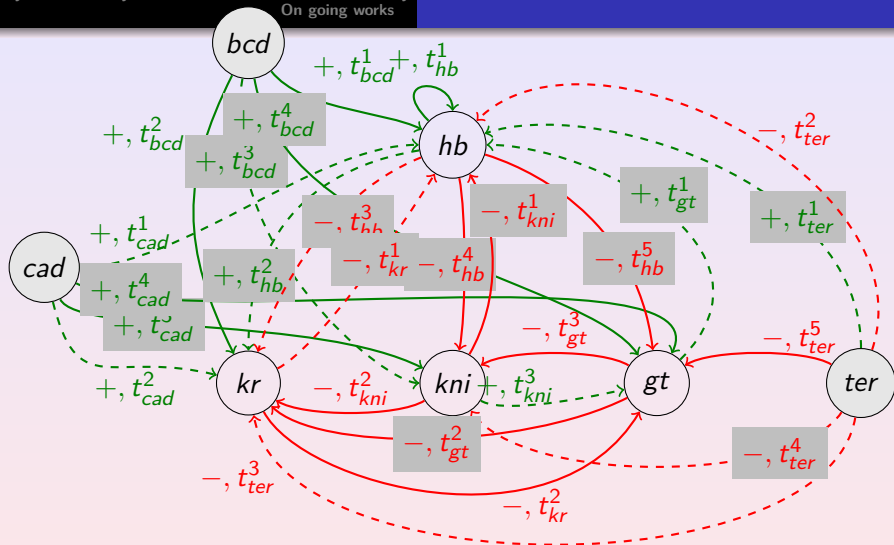
Automatic repairing, by relaxing as few as possible additivity constraints, offers two possibilities:

- Rejecting $K_{gyrAB} \geq K_{gyrAB}^{fis}$: means that *Fis* does not inhibit *GyrAB* when the bacteria are not stressed. **Disagrees** experimental data from [Schneider et al., Mol. Microbiol., 1999].

- Rejecting $K_{topA}^{fis} \geq K_{topA}$: would imply that TopA synthesis is possible even if the concentration of *Fis* is low. **Supported** by [Westein Fischer et al., Mol. Microbiol., 2007] for a stress due to hydrogen.

  ● By relaxing this last constraint, we get only 3 different instantiated models (on the 279,936 possible instantiated models due to the possible values of the 22 logical parameters).

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Drosophila embryo gap genes net. [Corblin et al. IPCAT 2012]

- Three maternal genes (cad, bcd, ter) and four gap genes (kr, hb, kni, gt). Well-established or **potential** interactions.
- Spatio-temporal expression profile of the main genes along the antero-posterior axis, giving seven regions (stable states).
- Expressions of genes also available from the seven **mutants**.

Objective: networks with the number of potential interactions and the number of thresholds **minimized**.
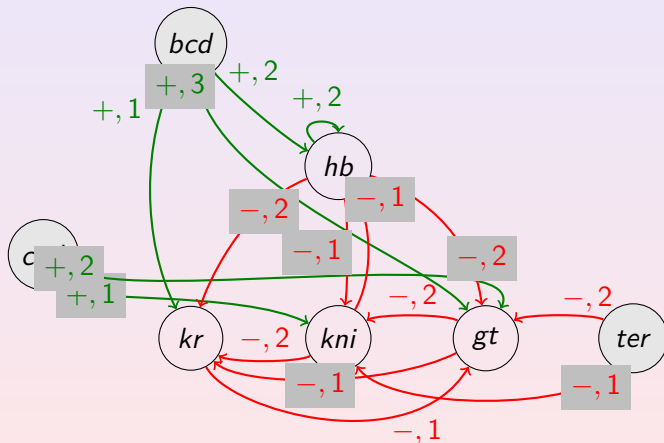
Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Potential interactions are represented by dotted red arrows.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Mutants and minimization. Results

- The set of constraints is found **consistent** (3338 s., on a PC, 2 proc. 2.4 GHz, 2.9GB memory).

- A **unique** minimal regulatory graph is then obtained (1016 s.) which includes only **two** potential interactions. Finally we get a unique minimal instantiation of the thresholds (368 s.).

- Deduced properties on parameters: 52 fixed parameters(over 72), 48 inequalities on the remaining ones: 12 between one parameter and one threshold, 36 between two parameters.

- The story is not finished... Enforcing CTL AF-like formulas is now required.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Minimized network, with only two potential interactions

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

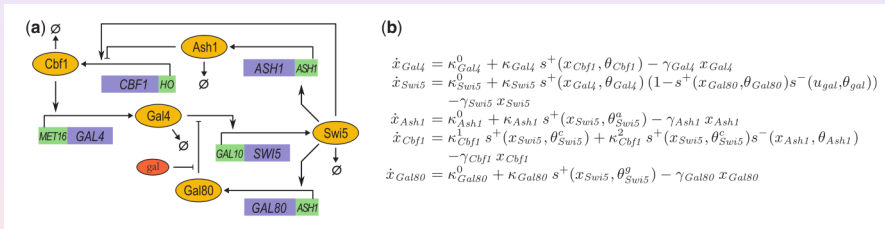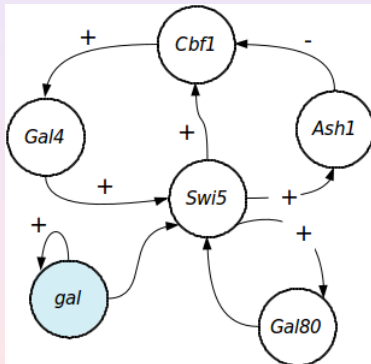# IRMA network, from [G. Batt et al., Bioinformatics, 2010]

Figure : (a) The IRMA (In vivo benchmarking of Reverse-engineering and Modeling Approaches) network [I.Cantone et al., Cell., 2009], (b) the corresponding piecewise affine differential equations.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

IRMA interaction network

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Temporal profiles

(a) Temporal profiles encoding of averaged gene expression.

(b) $\phi_1 \triangleq \mathbf{EX}(u_{gal}low \wedge \mathbf{EF}(B_{off} \wedge \mathbf{EF}(C_{off} \wedge \mathbf{EF}(\ldots)))) \wedge \mathbf{EX}(u_{gal}high \wedge \mathbf{EF}(B_{on} \wedge \mathbf{EF}(C_{on} \wedge \mathbf{EF}(\ldots))))$
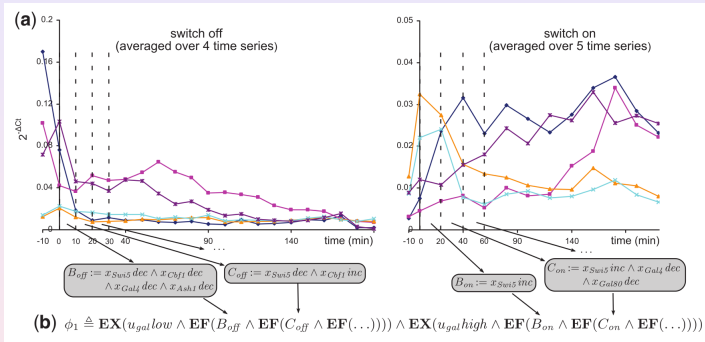
Figure : (a)Temporal profiles encoding of averaged gene expression.
"switch-on" ("swith-off") refers to the activation(inhibition) of Swi5 during
growth of galactose (glucose). (b) Temporal encoding of the switch-on
and switch-off behaviors. Only changes greater than 5×10^{-3} units are
considered significant

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Two approaches

- Batt's approach (with singular states): a new modeling with more regular states, such that each species has a unique derivative sign in them + the model checking tool NuSMV. He claims, when comparing its work, that it applies to "incompletely instantiated models" and provides "more precise results" and "efficient coding".

- Our approach for replying : Thomas initial model (no singular states), with adequate constraints expressing that a path satisfies a temporal series.

- Programming experiments for inferring parameters and thresholds : which models satisfy formulas $\varphi 1$ (2 large EF formulas representing averaged "switch-off" and switch-on" experiments) and $\varphi 2$ (9 large EF formulas representing all time series).

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Results

- After discovering mistakes in Batt's work (and Batt in both our works...), we found the <u>same</u> results: 64 models for $\varphi 1$ and 4 for $\varphi 2$ (on 4860). No more precise results with singular states...
- <u>Better or equivalent</u> performances: 139 s. vs 885 s. for $\varphi 1$, 2002 s. vs 2021 s. for $\varphi 2$, with a regular ASP solver (no incremental solving).

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Constraints vs model checking

- Model checking is based on CTL with a weak expression power compared to Prolog like ASP language, e.g. enforcing the existence of at least two steady states is not possible.
- Model checking is oriented toward verification of transition systems, Logic programming toward programming with logic, e.g. Batt performs out of NuSMV by using a counter-example delivery facility and a para-logic help for producing models by extension (not appropriate to deduce properties).

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency r
On going works

Content

1. Basics: ASP and logical modeling of Thomas GRNs

2. Applications

3. Generally true Additivity Constraints and automatic consistency repairing

4. On going works

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Two modeling issues

- Escaping inconsistency if some additivity constraints cannot not be satisfied because contraindications due to behaviors .
- Accepting only models with as many as "possible" additivity constraints.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

A first solution

- Enumerating all models, i.e. all possible atoms kparam(K, Ik) where K is the value of the parameter named Ik. **Costly**.
- Maximizing, with a para-logical process (Max-SAT like), the number of satisfied additivity constraints. **Costly** too.
  And debatable : **logical** minimization provided by stable models vs **para-logic** global criterion.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

New solution (main lines)

Firm production of kparam atoms restricted to:

- rules specifying the paths.
- rules specifying the observability constraints.
  Important: these constraints are disjunctions, like
  $(K_b < K_b^a) \lor (K_b^b < K_b^{ab})$. Non minimal models of them
  should be rejected (unless a contraindication due to a
  behavior), e.g. the rules must reject $(K_b < K_b^a) \land (K_b^b < K_b^{ab})$
  if possible. Naturally expressible in ASP.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency r
On going works

New solution. Efficient inconsistency repairing

Conditional production of kparam atoms due to the additivity constraints, by **default** rules like:

addit(+, a, b) :- obs(+, a, b), not obs(-, a, b).

where addit(+, a, b) implies kparam atoms satisfying $(K_b \leq K_b^a) \wedge (K_b^b \leq K_b^{ab})$.

In case of a negative observability due to some behavior, no inconsistency appears since the rule is not applicable.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

New solution. Retaining only appropriate models

- **Aim**: retaining logically models with maximum allowed additivity constraints.
- **Issue**: avoiding that additivity on one edge infers non additivity on another one (i.e. inappropriate influence between defaults).
- The following rules mimic such influences with op_ad1 interpreted as "additivity impossible for the interaction 1" (| stands for minimal disjunction):

  op_ad2 | ad1 :- not op_ad1.
  op_ad1 | ad2 :- not op_ad2.

  Three Anwer Sets (ASs) : {ad1,ad2}, {op_ad1} and {op_ad2}.

  Challenge : transform these rules to obtain a conjunction of defaults, i.e. with the only AS {ad1,ad2}.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Construction of conjunction of defaults

Two steps :

- Defining the literal c by :
  c :- op_ad1.
  c :- op_ad2.
  so that not c means that both op_ad1 and op_ad2 are
  unknown or false,
- Adding to each rule a new and "curious" (ASP) tautological
  term serving as a **guard**:
  op_ad2 | ad1 :- not op_ad1, 1{c, not c}1.
  op_ad1 | ad2 :- not op_ad2, 1{c, not c}1.
- Then we get only the AS {ad1,ad2}.
  If we add the rule : 1{op_ad1}1., we get both {ad1,ad2}
  and {op_ad1}. Then a para-logical maximization could be
  applied if wished.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Possible inconsistency. Example

Unfortunately, asserting all guards may lead to inconsistency.
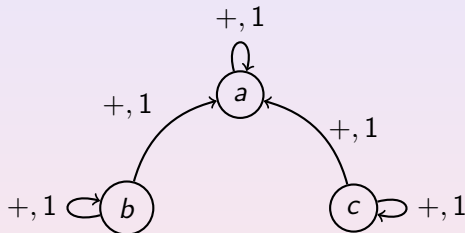
Observability constraints

$$(K_a < K_a^a \wedge K_a^a \geq 1) \vee (K_a^b < K_a^{ab} \wedge K_a^{ab} \geq 1)$$
$$\vee (K_a^c < K_a^{ac} \wedge K_a^{ac} \geq 1) \vee (K_a^{bc} < K_a^{abc} \wedge K_a^{abc} \geq 1)$$

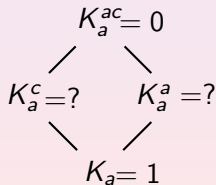$$(K_a < K_a^b) \vee (K_a^a < K_a^{ab}) \vee (K_a^c < K_a^{bc}) \vee (K_a^{ac} < K_a^{abc})$$

$$(K_a < K_a^c) \vee (K_a^a < K_a^{ac}) \vee (K_a^b < K_a^{bc}) \vee (K_a^{ab} < K_a^{abc})$$

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency r
On going works

Possible inconsistency (cont.)

The instantiations $K_a = 1$, $K_a^b = 0$, $K_a^{ab} = 0$, $K_a^{ac} = 0$, $K_a^{bc} = 0$
and $K_a^{abc} = 1$ ensure the observability constraints, but forbid both
additivity constraints related to the edges $a \rightarrow a$ and $c \rightarrow a$:
K_a^a and K_a^c, not a priori known, should be higher (resp. lower) than
or equal to $K_a = 1$ (resp. $K_a^{ac} = 0$).
See the lattice:

$$K_a^{ac} = 0$$

$$K_a^c = ? \qquad K_a^a = ?$$

$$K_a = 1$$

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Inconsistency prevention

Briefly

- For each couple of edges targeting a species N, constructing a guard guarantying the conjunction of their additivities only in case of absence of lattices like above (in a generalized form).
- Expressing the conjunction of all additivies on N as the logical conjunction of the above guards devoted to each couple of edges.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Some results

For the example above:

- In case no added extra-behavior: 101 ASs respecting only the observability constraints. With additivity constraints, reduction to 51 ASs with unitary defaults and to 9 ASs with the global guard defined above.

- With the extra-behavior $K_a = 0$, $K_a^a = 0$, $K_a^b = 0$, $K_a^c = 1$, $K_a^{ab} = 1$, still allowing models with all additivity constraints : 8 ASs with only observability constraints. With additivity constraints, reduction to 6 ASs with unitary defaults and to 1 AS with the global guard defined above.

- If adding $K_a^{ac} = 0$, that forbids the conjunction of defaults involving $a \to a$, one AS is (fortunately) obtained (non monotonic effect).

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Content

1. Basics: ASP and logical modeling of Thomas GRNs

2. Applications

3. Generally true Additivity Constraints and automatic consistency repairing

4. On going works

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

On going work. Ensuring CTL formulas in ASP

EF formulas, already available, for **analysis** purpose.
AF formulas necessary for **synthesis** purpose.
Trivial definitions (apparently):

- eF(Prop,S) :- ap(Prop,S).
  eF(Prop,S) :- not ap(Prop,S),
                transition(S,Sp), eF(Prop,Sp).
- aF(Prop,S) :- ap(Prop,S).
  aF(Prop,S) :- not ap(Prop,S),
                aF(Prop,Sp): transition(S,Sp).

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency r
On going works

Ensuring CTL formulas in ASP. Issues

- Checking loops, like :
  eF(Prop,s1) :- transition(s1,s2), eF(Prop,s2).
  eF(Prop,s2) :- transition(s2,s1), eF(Prop,s1).
  Given for free, thanks to the **minimality** of stable models !
- Actual work: implementation considering a limitation on the number of states instead of a limitation on the length of paths (adequate for EF formulas but not for AF formulas).

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

On going work. Multiplexes

When working with a Phd student (lab. IRcyn, Nantes) for comparison with another approach, we had to implement networks where multiplexes were present: ERBB receptor-regulated G1/S transition network (Sahin *et al.*, 18 species), tail resorption during the metamorphosis of tadpole (Khalis *et al.*, 8 species) and the T-cell Signaling network (Klamt *et al.*, 40 species)

Actual work: for efficiency and learning purpose, implementation of multiplexes similar to R. Thomas' SOP (Sum Of Products, disjunctive normal form) with a definition in terms of kinetic parameters.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

Multiplexes. Specification

- **Syntax** of the language of the interactions targeting a species
  x:
  $I ::= Mul \mid Mul$ **or** I
  $Mul ::= lu \mid lu$ **and** Mul
  $lu ::= Gene\_id \mid Sig\ Gene\_id$
  $Sig ::= + \mid -$
- **Semantics** : the value of $oc(I)$, a logical function of the
  parameters, constructed following the *composition principle*.
  Examples:
  $oc(+a) = K_x^a > K_x.$
  $oc(+a$ **and** $+b) = (K_x^{ab} > K_x) \wedge (K_x^a = K_x^b = K_x)$
  $oc(+a$ **or** $+b) = (K_x^{ab} > K_x^b \vee K_x^a > K_x) \wedge (K_x^{ab} > K_x^a \vee K_x^b > K_x)$
  $oc(-c$ **and** $+a$ **and** $+b) = (K_x^{ab} > K_x^c)$
  $$\wedge (K_x^{abc} = K_x^{ac} = K_x^{bc} = K_x^a = K_x^b = K_x = K_x^c)$$

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency
On going works

On going work. Mamalian circadian cycle

Knowing the existence of three such cycles (equinox, winter, summer), find models and delays. With a very reduced network (3 species including light). Delay modeling is rather simple, but demands at least large integers for expressing ratios of clocks. Actual work : implementation with **linear equations** on integers, provided by the ASP solver clingo.

Basics: ASP and logical modeling of Thomas GRNs
Applications
Generally true Additivity Constraints and automatic consistency r
On going works

Thanks

Thanks to Microsoft Research (scholarship for NM), ANR (Agence Nationale de la Recherche) projet CADMIA and the Postdam team (M. Gebser, T. Schaub) for the *clingo* ASP software.
And to Delphine, Emna, Claudine, ... Denis, Gilles, Gregory, Hans, Jean-Paul,...

Thank you for your attention

QUESTIONS ?

Declarative approach ...
Non monotonicity ...
Composition of defaults...

...